

FIG. 1

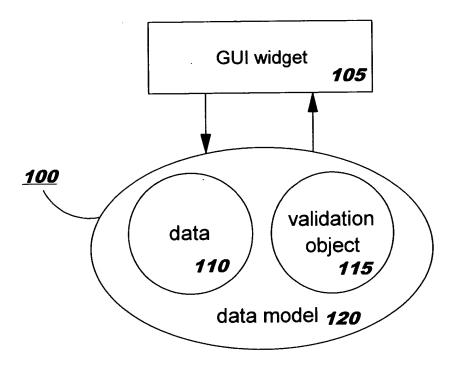


FIG. 2

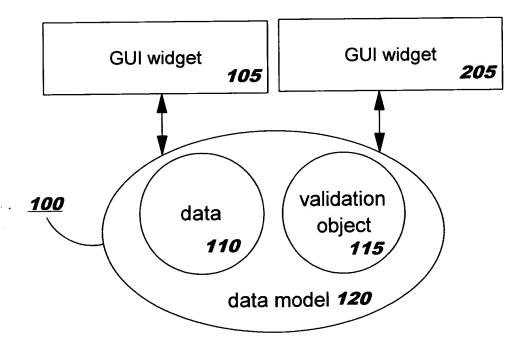




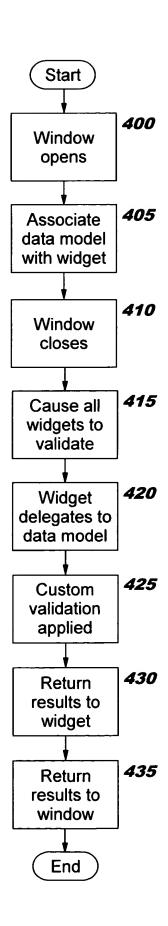
FIG. 3

300

```
305 < string Variable
      310 name="social_security_number"
      315 minimumLength="9"
      320 maximumLength="9">
      325 < labelText>SSN < / labelText>
      330 <helpText>The social security number of the applicant.</helpText>
      335 <mnemonic>D</mnemonic>
           <inputValidation>
            <ignore>
             <characters>-</characters>
      345
            </ignore>
340
            valid>
             <characters>0123456789-</characters>
      350
            </valid>
           </inputValidation>
          </stringVariable>
```



FIG. 4



}



FIG. 5A

```
public class StringVariableModel extends VariableModel
    A template method which calls subclass methods (which should
       call super.doValidate()), then employs the Strategy pattern by calling
       all of the set of CustomValidators. The first error causes the function
       to return. If there is no error, then all listeners will be notified
       of the new value.
 */
 public boolean validate(Object value, boolean displayErrors)
  String errorString;
  String strVal = (value == null) ? new String("") : value.toString();
  VariableChangedEvent vce = new VariableChangedEvent(this, strVal);
  errorString = doValidate(value, vce);
  if ((errorString == null) || (errorString.compareTo("") == 0))
   Iterator it = validators.iterator();
   while (it.hasNext())
     errorString = ((CustomValidator)it.next()).validate(value);
    if ((errorString!= null) && (errorString.compareTo("")!= 0))
      break;
   }
  if ((errorString != null) && (errorString.compareTo("") != 0))
   // put up a dialog or message box containing the error message.
   if (displayErrors)
     JOptionPane.showMessageDialog(null, errorString, getLabelText(),
        JOptionPane.ERROR MESSAGE);
   return false;
  processValueChanged(vce);
  return true;
```



FIG. 5B

```
public void addValidator(CustomValidator cv)
{
   validators.add(cv);
}

public void removeValidator(CustomValidator cv)
{
   validators.remove(cv);
}
```